IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent application of:
Fontoura et al.

Serial No.: 10/723,391

Filed: November 25, 2003

Group Art Unit: 2168

Examiner: Oni, Olubusola

Atty. Docket No.: ARC920030080US1

For: USING INTRA-DOCUMENT INDICES TO IMPROVE XQUERY PROCESSING OVER XML STREAMS

Commissioner of Patents
P.O. BOX 1450
Alexandria, VA 22313-1450

DECLARATION UNDER 37 C.F.R. §1.131

We, the inventors of the invention defined by claims 1-37 of U.S. Patent Application Serial No. 10/723,391 hereby declare the following:

[0001] The purpose of this declaration is to prove that we conceived the claimed invention prior to the earliest effective prior art date of U.S. Patent Publication No. 2005/0055343 published to Krishnamurthy, which is presently understood to be September 4, 2003. The following shows that we conceived our invention prior to September 4, 2003 and that we were diligent from our date of conception to its reduction to practice and were further diligent to the date of the filing of our patent application, which has a filing date of November 25, 2003 (hereinafter referred to as the "Patent Application").

[0002] We are all the inventors of the subject matter claimed in claims 1-37 of U.S. Patent Application Serial No. 10/723,391.

[0003] During all time periods mentioned herein, and specifically between our conception date and the filing date of the application, all activities described herein occurred in the United States.

[0004] Proof of the conception of the claimed invention prior to September 4, 2003, and diligence in reducing the invention to practice and filing the Patent Application is demonstrated in the attached Exhibits, labeled as Exhibit A and B.

[0005] As shown in Exhibit A, which is an invention disclosure form typically used by the designated Assignee, International Business Machines Corporation, we conceived the claimed invention at a date prior to September 4, 2003. As permitted by MPEP §715.07, the dates on Exhibit A have been removed; however, we hereby declare that all dates corresponding to the conception date and reduction to practice occurred prior to September 4, 2003. Further, the invention was actually conceived before Exhibit A was prepared. Therefore, our conception date actually predates Exhibit A.

[0006] Exhibit A specifically discloses the claimed invention as defined by the independent claims. For example, independent claim 1 defines a method for parsing documents in query processing, said method comprising producing at least one index of a document written in a mark-up language; corresponding said index to said document; scanning said document; and selectively skipping portions of said document based on instructions from said index. Independent claim 13 defines a system for parsing documents in query processing, said system comprising at least one index corresponding to a document written in a mark-up language; a processor operable for scanning said document; and a parser operable for selectively skipping portions of said document based on instructions from said index. Independent claim 25 defines a program storage device readable by computer, tangibly embodying a program of instructions executable by said computer to perform a method for parsing documents in query processing, said method comprising producing at least one index of a document written in a mark-up language; corresponding said index to said document; scanning said document; and selectively skipping portions of said document based on instructions from said index. Independent claim 37 defines a system for efficiently parsing documents in query processing, said system comprising means for producing at least one index of a document written in a mark-up language; means for corresponding said index to said document; means for scanning said document; and means for selectively skipping portions of said document based on instructions from said index.

[0007] Exhibit A clearly describes the above features (and in particular, the Background Section, Summary of Invention Section, and Description Section provided on pages 2-5 of Exhibit A). In fact, the descriptions provided in pages 2-5 of Exhibit A served as the basis for the specification, drawings, and claims of the Patent Application. The features provided in dependent claims 2-12, 14-24, and 26-36 are generally inferred in Exhibit A.

[0008] As shown in Exhibit B, which are notes taken during an invention review meeting that further identified aspects of the invention, we conceived the claimed invention at a date prior to September 4, 2003. As permitted by MPEP §715.07, the dates on Exhibit B have been removed; however, we hereby declare that all dates corresponding to the conception date and reduction to practice occurred prior to September 4, 2003. Further, the invention was actually conceived before Exhibit B was prepared. Therefore, our conception date actually predates Exhibit B.

[0009] Exhibit B specifically discloses the claimed invention as defined by the independent claims and in the features identified as claimable subject matter as provided on page 2 of Exhibit B. In fact, the features provided on pages 2 of Exhibit B served as the basis for the claims of the Patent Application. The features provided in dependent claims 2-12, 14-24, and 26-36 are generally inferred in Exhibit B.

[0010] We were diligent from the date of conception in reducing the invention to practice and in pursuing, preparing, and filing the Patent Application. More specifically, on August 29, 2003, information similar to that shown in Exhibits A and B were presented to a patent attorney to determine whether a patent application should be prepared.

[0011] Generally, the invention was conceived on or about March 1, 2003 and was reduced to practice on or about March 30, 2003. An exhaustive series of experiments were conducted on the invention testing its validity from March 1, 2003 to June 30, 2003. The testing was quite rigorous and required substantial time, money, and effort to undertake. The results of the experiments were positive, which further resolved the decision to seek patent protection.

After the invention was conceived and reduced to practice, and the testing yielded positive results, the decision was reached to seek patent protection due to the potential commercial value and prestige afforded by the claimed invention as well as the results of a prior art search. On August 29, 2003, a patent attorney was instructed to prepare a patent application that eventually became the Patent Application. The Patent Application was eventually prepared and filed on November 25, 2003.

[0012] The foregoing declarations are made according to our best recollection upon review of the appropriate documents and notes, and I hereby acknowledge that willful false statements and the like are punishable by fine or imprisonment, or both (18 USC §1001) and may jeopardize the validity of the application or any patent issuing thereon. All statements made herein are made of our own knowledge and are true and all statements that are made on information and belief are believed to be true.

Marcus F. Fontoura

7/28/2006
Date

Vanja Josifovski

7/31/2006
Date

Pratik Mukhopadhyay

Date

After the invention was conceived and reduced to practice, and the testing yielded positive results, the decision was reached to seek patent protection due to the potential commercial value and prestige afforded by the claimed invention as well as the results of a prior art search. On August 29, 2003, a patent attorney was instructed to prepare a patent application that eventually became the Patent Application. The Patent Application was eventually prepared and filed on November 25, 2003.

[0012] The foregoing declarations are made according to our best recollection upon review of the appropriate documents and notes, and I hereby acknowledge that willful false statements and the like are punishable by fine or imprisonment, or both (18 USC §1001) and may jeopardize the validity of the application or any patent issuing thereon. All statements made herein are made of our own knowledge and are true and all statements that are made on information and belief are believed to be true.

________________________ ______________
Marcus F. Fontoura Date

________________________ ______________
Vanja Josifovski Date

Pratik Mukhopadhyay 07-28-2006
Pratik Mukhopadhyay Date

EXHIBIT A

Disclosure ARC8-2003-0084

Prepared for and/or by an IBM Attorney - IBM Confidential

Created By Marcus Fontoura On [redacted] 09:56:56 AM MDT
Last Modified By Vanja Josifovski On [redacted] 12:11:28 PM EDT

Required fields are marked with the asterisk (*) and must be filled in to complete the form .

*Title of disclosure (in English)

Using intra-document indices to improve XQuery processing over XML streams

Summary

Status		Under Evaluation
Final Deadline		
Final Deadline Reason		
*Processing Location		Almaden
*Functional Area	select	(8CC) 8CC - Exploratory DB - (W.Cody)
Attorney/Patent Professional		Marc D McSwain/Almaden/IBM
IDT Team	select	Daniel M Shiffman/Almaden/IBM Marc D McSwain/Almaden/IBM Bill Cody/Almaden/IBM
Submitted Date		[redacted] 02:28:16 PM MDT
*Owning Division	select	RES
Incentive Program		
Lab		
*Technology Code		601
PVT Score		

Inventors with a Blue Pages entry

Inventors: Vanja Josifovski/Almaden/IBM@IBMUS, Marcus Fontoura/Almaden/IBM

Inventor Name	Inventor Serial	Div/Dept	Inventor Phone	Manager Name
Josifovski, Vanja	3A5212	22/K55I	457-1719	Cochrane, Roberta (Bobbie)
> Fontoura, Marcus F.	3A5041	22/8CCD	457-1416	Shekita, Eugene J.

> denotes primary contact

Inventors without a Blue Pages entry

	Pratik Mukhopadhyay
Serial Number :	(N/A)
Company :	University of Califonia at San Diego
Citizen of :	India
E-Mail :	
Business Address :	

rec'd [redacted]
(vacation)
asked Bill for a
asked about non-I
inventor.

Business Phone :
Home Address :

IDT Selection

Attorney/Patent Professional: Marc D McSwain/Almaden/IBM

IDT Team: Daniel M Shiffman/Almaden/IBM
Marc D McSwain/Almaden/IBM
Bill Cody/Almaden/IBM

Response Due to IP&L [redacted]

***Main Idea**

1. Background: What is the problem solved by your invention? Describe known solutions to this problem (if any). What are the drawbacks of such known solutions, or why is an additional solution required? Cite any relevant technical documents or references.

Most of the XPath and XQuery implementations today process queries by traversing an in-memory representation of the document using the Document Object Model (DOM) interface. In DOM at any point the processing can move in any direction in the XML tree from the current node to its children, its parent or any of its siblings. While this makes the implementation easier, the requirement that the whole document in memory is a major drawback of this approach, leading to large memory consumption (decreased concurrency) and high latency (the document needs to be processed before the first answer is produced). In order to overcome these limitations, streamed implementations based on the Simple API for XML (SAX) interface are emerging. At the Almaden Research Center we have developed the TurboXPath processor that can evaluate single-document XQuery queries over streams of XML data using SAX. TurboXPath has demonstrated to reduce both the memory consumption and the latency by orders of magnitude. Nevertheless experiments have demonstrated that XML parsing (producing SAX events from an XML document stream) is responsible for 60 to 95 percent of the overall processing time. This invention describes how

intra-document indices can be used to reduce parsing time in the context of processing XQuery queries over XML documents stored on disk and streamed into the system.

2. Summary of Invention: Briefly describe the core idea of your invention (saving the details for questions #3 below). Describe the advantage(s) of using your invention instead of the known solutions described above.

One of the reasons for the high overhead of the parsing is that the parsers produces events for all document pieces, regardless if they are relevant for processing the query. In this invention we propose an index that added to the XML documents aids the parser in:

1. Skipping pieces from the document and
2. Extracting result portions without first turning them into events and stringifying again.

3. Description: Describe how your invention works, and how it could be implemented, using text, diagrams and flow charts as appropriate.

This invention proposes an index that is added to a textual XML document or stream. As opposed to some binary XML representations that require modifications of the document format (such as XTalk), in the proposed approach the original document is left unchanged. This has three major advantages:

1) To extract a piece of the document, the processor does not need to recreate the result XML from the binary format. The index contains information that allows for efficient extraction of elements from the original document.

2) The size of the index can be controlled by indexing only parts of the index. In the non-indexed parts of the documents the processing would be the same as if there were no index, with no performance penalty. This is especially useful in scenarios that there is a known limit on the query depth, for instance.

3) This approach does not require any changes in parsers not supporting the use of the index. More efficient parsers can take advantage of this index and improve the performance. Traditional parsers will ignore the index.

The changes in the interface to the parser to allow the query processor or application to take advantage of the new features are very little. Three new functions are introduced to the SAX interface: skipElement(), skipAndSaveElement(), and getElementByteSize(). Function skipElement() is called within the startElement() SAX handler and instructs the parser to skip all events up to and including the end element event matching the currently processed startElement() event. The skipAndSaveElement() is similar to skipElement() except that it stores the textual content of the element into the provided buffer. The size of element is obtained using the third function, getElementByteSize(). All of these operations are efficiently implemented using the index as described below.

The index structure was designed to allow the TurboXPath and other query processors to skip over fragments of the documents which were not relevant to the query being evaluated. The index structure contains the end position and the number of subelements of each element in the document. The order of the entries in the index correspond to the order of the elements in the document to allow the application (TurboXPath) to traverse the index in lock step with the input document. Every time TurboXpath receives a start event, it advances the current position in the index. While processing the start event, if the application decides that the rest of this element can be skipped, it uses the information about the end position of the current element which is available in the current index entry to determine the position (in the input document) where the parser should resume scanning the input. The information in the current index entry about the number of subelements of the current element is used to update the current position in the index. If the index entry indicated the current element had k subelements, the current position in the index is advanced k positions. This step keeps the current position in the document and the index synchronized from the applications viewpoint. An example of the index is shown in figure below, for a sample XML document from the DBLP publication database.

XML document:

```
<dblp>
        <proceedings key="conf/vldb/2000">
                <editor>Amr El Abbadi</editor>
                <editor>Michael L. Brodie</editor>
                <editor>Sharma Chakravarthy</editor>
                <editor>Umeshwar Dayal</editor>
                <editor>Nabil Kamel</editor>
                <editor>Gunter Schlageter</editor>
                <editor>Kyu-Young Whang</editor>
                <title>VLDB 2000, Proceedings of 26th International Conference on Very Large Data
Bases, September 10-14, 2000, Cairo, Egypt</title>
                <publisher href="db/publishers/mkp.html">Morgan Kaufmann</publisher>
                <year>2000</year>
                <isbn>1-55860-715-3</isbn>
                <url>db/conf/vldb/vldb2000.html</url>
        </proceedings>
        <proceedings key="conf/vldb/2001">
                <editor>Peter M. G. Apers</editor>
                <editor>Paolo Atzeni</editor>
                <editor>Stefano Ceri</editor>
                <editor>Stefano Paraboschi</editor>
                <editor>Kotagiri Ramamohanarao</editor>
                <editor>Richard T. Snodgrass</editor>
                <title>VLDB 2001, Proceedings of 27th International Conference on Very Large Data
Bases, September 11-14, 2001, Roma, Italy</title>
                <publisher href="db/publishers/mkp.html">Morgan Kaufmann</publisher>
                <year>2001</year>
                <isbn>1-55860-804-4</isbn>
                <url>db/conf/vldb/vldb2001.html</url>
        </proceedings>
</dblp>
```

Index:

ELEMENT	END POSITION	NUMBER OF CHILDREN
<dblp>	1119	25
<proceedings>	571	12
<editor>	73	0
<editor>	107	0
<editor>	144	0
<editor>	176	0
<editor>	204	0
<editor>	239	0
<editor>	272	0
<title>	404	0
<publisher>	473	0
<year>	491	0
<isbn>	519	0
<url>	557	0
<proceedings>	1113	11
<editor>	642	0
<editor>	672	0
<editor>	702	0
<editor>	738	0
<editor>	777	0
<editor>	815	0
<title>	947	0
<publisher>	1016	0
<year>	1034	0
<isbn>	1060	0
<url>	1098	0

In order to clarify the presentation we used the tag name in the ELEMENT column of the index but in the implementation tag IDs can be used. Let us now consider how the index could be used to enhance the processing of the query:

dblp/proceedings[@key = "conf/vldb/2000"]//editor

Processing would proceed normally for the first of the two <proceedings> entries until the first <title> subelement is found. All the <editor> subelements would match the query and would be returned to the user. In the start element of <title>, TurboXPath would use the index and decide to skip that element, jumping to position 404 in the document and to the next entry of the index. The same would happen for the next four subelements: <publisher>, <year>, <isbn>, and <url>. The second <proceedings> element would be completely skipped, since it does not match the query. Consulting the index in the start element event of <proceedings key="conf/vldb/2001">, TurboXPath would skip to document offset 1113 and it would also skip the next 11 entries of the index.

In order to control the index size the application may decide not to index certain portions of the document. In this example, if the application decides not to index the <proceedings> subelements the index would be:

ELEMENT	END POSITION	NUMBER OF CHILDREN
<dblp>	1119	2
<proceedings>	571	0
<proceedings>	1113	0

This is a much more compact index that still allows big jumps (and big performance improvements) for several queries. For our sample query this index not allow the skipping of the non-matching subelements of the first <proceedings> entry but it would still allow the application to skip the second <proceedings> entry completely.

***Patent Value Tool**

* **1.** Select the single most appropriate technology category for your invention from the following technologies list.
(601) PPM 600 Software/Services/ Applications/Solutions-601 Database programs
Comments
Are there any additional significant markets where the invention is likely to have impact?
● Yes ○ No

Please identify them:
Life sciences

*2. Have you implemented the invention (e.g., made a prototype) or otherwise shown that it is workable?
● Yes ○ No

*3. Has the subject matter of the invention or a product incorporating the invention been offered for sale, or is it likely to be offered for sale, as part of an IBM product or service?
○ No known product plans within 2 years
● Maybe; GA 1-2 years away
○ Yes; GA within 3-12 months
○ Yes; GA within 3 months
○ Yes; product has been announced

What product?
Trevi, DB2

What is the significance of the invention within the product?
● Improves general usability
○ Enables a minor feature
○ Enables a major feature

What feature?
XML processing

*4. Has the invention been commercially used (internally or externally) by IBM or another entity (e.g., included in or used to make products, or prototypes provided to a customer)?
○ Yes ● No

EXHIBIT B

Notes for ARC8-2003-0084

Review meeting [redacted]

Xpath, Xquery - prior art uses DOM interface to traverse in-memory document representation; memory and latency problems. SAX for streaming document helps, but XML parsing still takes 60-95% of the time, so problem remains.

Present invention uses intra-document indices to reduce parsing time when processing Xquery queries over XML documents (preferably streaming documents). Prior art parsers produce events for all document pieces, regardless of query relevance in the prior art. So, the present invention adds an index to XML documents to help the parser (1) skip pieces and (2) extract result portions without first turning them into events and stringifying again. The original document is left unchanged (i.e. no modification to the document format).

Advantages:
1. Index allows efficient extraction without recreating result XML from binary format (indexing into binary stream?)
2. Size of index is controlled - only parts of documents need be indexed. Non-indexed document portions can be processed as usual.
3. Regular parsers still work - they just ignore any index that might be there (backwards compatible)

Functions added to parser:
1. skipElement = skip all events up to and including end element event
2. skipandsave = ditto, but saves text into a buffer
3. getsize = determines size of element

Index lists element, end position, number of children (subelements)
Not all elements need be indexed, if not needed for a query.
Incorporate CHA9-2003-0002-US1 by reference
No bar date - no product ship date or publication date

Claims:

A method for efficiently parsing documents, comprising:
-producing at least one index of a document written in a markup language
-adding said index to said document
-selectively skipping portions of said document

deps on:
-markup language is HTML
-markup language is XML
-no reformatting of document needed to add index (in contrast with Xtalk)
-index contents include at least one of (element, end position, number of children)
-index may be limited, according to query relevance
-skipping done according to query relevance
-can create index a priori or can index by query history or probable query pattern
-can have more than one index, select by query relevance
--index is per document, could be large, so might be advantageous to have many smaller indices
-document can be streamed, or not. Primarily for streaming
-one-pass algorithm
-discoverable by just changing the index - easy to detect use
-SAX interface is standard so if this invention is eventually to be submitted as part of a standard then Gerald Lane must be involved
CHA9-2003-0002 buffers streamed fragments that meet an evaluation criteria (e.g. relevant to query), so just saving portions of document is known

Claims:

A method for efficiently parsing documents, comprising:
-producing at least one index of a document written in a markup language
-adding said index to said document
-selectively skipping portions of said document

deps on:
-markup language is HTML
-markup language is XML
-no reformatting of document needed to add index (in contrast with Xtalk)
-index contents include at least one of (element, end position, number of children)
-index may be limited, according to query relevance
-skipping done according to query relevance
-can create index a priori or can index by query history or probable query pattern
-can have more than one index, select by query relevance
--index is per document, could be large, so might be advantageous to have many smaller indices
-document can be streamed, or not. Primarily for streaming
-one-pass algorithm
-discoverable by just changing the index - easy to detect use
-SAX interface is standard so if this invention is eventually to be submitted as part of a standard then Gerald Lane must be involved
CHA9-2003-0002 buffers streamed fragments that meet an evaluation criteria (e.g. relevant to query), so just saving portions of document is known